

### REMARKS

The present amendment is in response to the Office Action dated October 6, 2003. Claims 1-17 are now present in this case.

Claims 1-17 stand rejected under 35 U.S.C. § 102(e) as anticipated by U.S. Patent No. 6,529,208 B1 to Chun et al. The applicants respectfully disagree with the assessment of Chun and its applicability to the claimed invention.

### The Invention

The present invention is may be best understood with respect to Figure 32 and the accompanying description on pages 34-36 of the specification. The present invention is embodying in a pixel mask software module. In an exemplary embodiment, the pixel mask software module is located intermediate the graphics display interface (GDI) and the display driver. In a conventional computer system, the operating system, or an application executing thereon, delivers data to the GDI processes the data appropriately and, in turn, delivers the data to the display driver. In a typical embodiment, the display driver is a software driver that may be unique to the specific hardware graphics display adapter plugged into the computer. If multiple applications are running on the computer, the operating system and the GDI allocate the display space. For example, an application in which a popup menu appears is still under control of the operating system and functions through the GDI.

With the introduction of the pixel mask software intermediate the GDI and the display driver, a software application may have an application program interface (API) that operates independently of the operating system to specify a region of the display to be reserved for use with that application. The data from the API is delivered to the pixel mask software. The pixel mask software determines that the data is delivered from an authorized source associated with the reserved region of the display and clips data from operating system programs delivered via the GDI. The data from the GDI, and the altered portion are delivered to the display driver for processing in a conventional manner.

The location of the pixel mask software in this particular location provides relative platform independence and does not require customized driver software for each different type of hardware graphics display adapter. As will be discussed in greater detail below, Chun is directed to a modified hardware graphics display adapter that provides an overlay buffer that can be logically combined with other image data.

### Discussion of Rejections

With respect to claim 1, the Office Action states that Chun discloses a system in which a display mask region is defined on a display area of the video display system and associated with an authorized display source and, upon receipt of an indication from the authorized display source writing the image within an area defined by the display mask region to transparently write the image on the display area such that output from an unauthorized source is not displayed within the area all under control of code that is independent of a native operating system. This is an incorrect assessment of Chun.

Chun discloses a hardware device (see Figure 6) that accepts data from an operating system and is by no means independent of the operating system. Chun merely discloses a technique for rendering pixels from two frame buffers using only a single windows identification (WID) buffer. (See column 5, lines 22-25.) As described in Chun, overlay buffers are logically ORed together by the hardware interface. (See column 5, lines 32-33.) However, all data for the buffers are provided by, and under control of, the native operating system. The fact that one buffer may overwrite another buffer when the logical OR operation occurs does not infer that the overlay buffer is provided with data independent of the native operating system. Furthermore, there is no suggestion that a display region mask is generated independent of the native operating system, as specified by claim 1.

Furthermore, claim 1 recites "associating the generated display region mask with the authorized display source" which must also occur under code that is independent of a native operating system. The Office Action states that such association is taught by Chun associating a display mask with a WID. However, it is

clear that WID association occurs under control of the operating system. Chun provides no teaching or even suggestion of any code independent of a native operating system that specified such WID. Indeed, Chun does not use the term "authorized display source" and provides no suggestion for any source of display data other than the windows operating system providing data to the hardware device described in Chun.

Finally, claim 1 recites "upon receiving an indication from the authorized display source to write the image within the area defined by the associated display region mask, transparently writing the image onto the display are, such that output from an unauthorized source is not displayed within the area defined by the associated display mask region in a manner that is independent of any display ordering imposed by the native operating system." As noted above, Chun does not teach or suggest any association of a display region mask with an authorized display source, and thus has no need to receive an indication from an authorized display source. Chun teaches providing display data from the operating system to the hardware device in multiple buffers and then providing a technique for overlaying the buffers in a hardware implementation. Clearly, the data for the various buffers in Chun comes from the operating system. There is no suggestion of any technique under control of code that is independent of the native operating system in which an authorized display source provides an indication to write the image within an area defined by an associated display region mask, as recited by claim 1.

Chun is a hardware implementation in which the data for the multiple buffers (e.g., the color buffer and the overlay buffer) are provided by the operating system. Chun provides no teaching or suggestion of any data or code that operates independently of an operating system. The hardware implementation of Chun merely provides a hardware buffer overlay technique in which data provided by the operating system into the overlay buffer is logically combined with other data provided by the operating system. Chun would also require a specialized display driver to accommodate the custom hardware. Such an approach is disadvantageous because it requires specialized hardware (*i.e.*, the graphics display adapter) and software (*i.e.*, the

display driver). Claim 1 is clearly directed to a method operating independently of the native operating system and allows an image provided by an authorized display source associated with a generated display region mask (under control of code that is independent of a native operating system) to the display area. Chun does not teach or suggest such a method. Accordingly, claim 1 is clearly allowable over Chun.

Claim 2 is directed to a method of preventing a first application from overwriting data displayed by a second application on a video display system. Claim 2 recites *inter alia* "generating a display region mask that defines a display area of the video display system" as well as "associating the generated display region mask with the second application." Claim 2 further recites "receiving data for the first application from a graphics device interface associated with a native operating system" and "modifying a portion of the received data intended for the display area defined by the display region mask to prevent the data from the first application from being displayed in the display area defined by the display region mask." Finally, claim 2 recites "transferring the data, including the modified portion, to a display driver associated with the video display system." As discussed above, this implementation may be more easily understood with respect to the exemplary embodiment in Figure 32, which is discussed on pages 34-36 of the application. As readily seen in Figure 32, the inventive method operates at a level between the graphics display interface and a display driver. Chun teaches directly away from such an implementation. Claim 2 recites a method in which data from a first application is received from a graphics device interface associated with a native operating system modified and transferred to a display driver associated with the video display system. In sharp contrast, Chun is a hardware device in which multiple sets of data (e.g., color data and overlay buffer data) are provided to a hardware device (i.e., a RAMDAC 17 stage pipeline 604) along with data provided by a WID buffer. The output of the RAMDAC 604 is sent to a display device. (See column 4, lines 55-56.) Thus, Chun describes a special hardware graphics adapter, which teaches directly away from a method in which data is received from a graphics device interface, modified, and transferred to a display driver, as recited by claim 2. Thus, claim 2 is clearly allowable over Chun. Claims 3-7 are also

allowable in view of the fact that they depend from claim 2, and further in view of the recitation in each of those claims.

Claim 8 is a system claim and recites *inter alia* "a display filter to intercept function calls from a graphics device interface associated with a native operating system." When a display filter detects that an intercepted function call from a first application is specifying transmission of data to a masked display area, the display filter operates to "clip a portion of the received data intended for the masked display are to prevent the data from the first application from being displayed in the masked display are."

As noted above, Chun is directed to a hardware graphics display adapter that receives its data from the graphics display interface of a native operating system. Chun provides no teaching or suggestion for intercepting function calls from a graphics device interface. All data in Chun, including the overlay buffer, is provided by the operating system via the graphics device interface. This is significantly different from the system of claim 8 in which a display filter intercepts function calls from the graphics device interface and, when the display filter detects that the intercepted function call is transmitting data to the marked display area, clipping a portion of the received data to prevent data from the first application from being displayed in the marked display area. This is significantly different from the hardware overlay buffer approach described by Chun. Chun does not teach or suggest the structures of Figure 7. Indeed, Chun teaches away from a device operating to intercept graphic display interface function calls by disclosing a hardware implementation of a graphics display adapter. Accordingly, claim 8 is clearly allowable over Chun. Claims 9-11 are also allowable in view of the fact that they depend from claim 8, and further in view of the recitation in each of those claims.

Claim 12 is a computer readable media claim containing instructions for controlling a computer processor to prevent a first application from overwriting data displayed by a second application on a video display system. Claim 12 recites *inter alia* that the computer instructions control the computer processor to receive "data for the first application from a graphics device interface associated with a native operating

system" as well as "clipping a portion of the received data for the display area defined by the display region mask to prevent the data from the first application from being displayed in the display area defined by the display region mask."

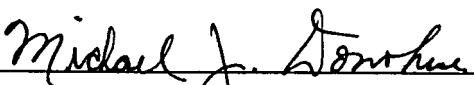
As discussed above, Chun teaches directly away from the claimed invention by disclosing a hardware graphics display adapter that operates at a totally different location within a computer. That is, the invention of Chun operates at the hardware interface level under control of the operating system. Data from the graphics display interface and display driver are provided to the various buffers in Chun and the modified output (*i.e.*, the logically ORed data and overlay buffer) are provided to the display screen. In sharp contrast, the computer readable media of claim 12 receives data from the graphics device interface associated with the native operating system and modifies that data by "clipping a portion of the received data and send it for the display area defined by the display region mask." Thus, claim 12 is clearly allowable over Chun. Claims 13-17 are also allowable in view of the fact they depend from claim 12, and further in view of the recitation in each of those claims.

The applicants have made a good faith effort to place all claims in condition for allowance. In view of the above remarks, reconsideration of the subject application and its allowance are kindly requested. If questions remain regarding the present application, the Examiner is invited to contact the undersigned at (206) 628-7640.

Respectfully submitted,

D. David Nason et al.

Davis Wright Tremaine LLP

  
\_\_\_\_\_  
Michael J. Donohue  
Registration No. 35,859

MJD:gatc  
2600 Century Square  
1501 Fourth Avenue  
Seattle, Washington 98101-1688  
Phone: (206) 622-3150  
Fax: (206) 6628-7699  
1420724\_1.DOC 59312-58